

Assignment of Cooperating UAVs to Simultaneous Tasks using Genetic Algorithms

Tal Shima* and Corey Schumacher†

Air Force Research Labs, Wright-Patterson AFB, OH, 45433

A problem of assigning multiple unmanned aerial vehicles (UAVs) to simultaneously perform cooperative tasks on consecutive targets is posed as a new NP-hard combinatorial optimization problem. The investigated scenario consists of multiple ground moving targets prosecuted by a team of heterogeneous UAVs carrying designated sensors and/or weapons. To successfully prosecute each target it first needs to be simultaneously tracked by multiple UAVs, from significantly different line of sight angles to reduce the position estimate errors, and then attacked by a different UAV carrying a weapon. Even for small sized scenarios, the problem has prohibitive computational complexity for classical combinatorial optimization methods due to timing constraints on the simultaneous tasks and the coupling between task assignment and path planning for each UAV. A genetic algorithm (GA) is proposed for efficiently searching the space of feasible solutions. A matrix representation of the GA chromosomes simplifies the encoding process and the application of the genetic operators. To further simplify the encoding, the chromosome is composed of sets of multiple genes, each corresponding to the entire set of assignments on each target. Simulation results confirm the viability of the proposed assignment algorithm for different sized scenarios. The sensitivity of the performance to variations in GA tuning parameters is also investigated.

Nomenclature

C	set of genes for each chromosome
f	fitness function
G	set of ground targets
J	cost function
J_1	cost of best assignment from the set of N_f initial random solutions
N_c	number of genes in each chromosome
N_f	number of chromosomes in a GA population
n_f	upper bound on number of feasible assignments
N_g	number of generation a GA is run
N_s	number of assignment stages
N_t	number of ground targets
N_v	number of stand-in UAVs
p_c	crossover probability
p_m	mutation probability
r_t	radius of stand-in UAV circular tracking trajectory
r_{max}	maximum range of stand-in UAV sensor
R_{min}	minimum turn radius of the stand-in UAV
r_{min}	minimum range of stand-in UAV sensor
S	set of assignment stages
t_{flight}	weapon minimum flight time

*This research was performed while the author held a National Research Council Research Associateship Award at the Control Design and Analysis Branch, Air Vehicles Directorate, Room 304, Building 146; Senior member AIAA; shima_tal@yahoo.com

†Research Aerospace Engineer, Control Design and Analysis Branch, Air Vehicles Directorate, Room 305, Building 146; Senior member AIAA; Corey.Schumacher@wpafb.af.mil

Report Documentation Page			<i>Form Approved OMB No. 0704-0188</i>	
<p>Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.</p>				
1. REPORT DATE AUG 2005	2. REPORT TYPE	3. DATES COVERED 00-00-2005 to 00-00-2005		
4. TITLE AND SUBTITLE Assignment of Cooperating UAVs to Simultaneous Tasks using Genetic Algorithms			5a. CONTRACT NUMBER	
			5b. GRANT NUMBER	
			5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S)			5d. PROJECT NUMBER	
			5e. TASK NUMBER	
			5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Air Force Research Laboratory, Air Vehicles Directorate, Wright Patterson AFB, OH, 45433			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)			10. SPONSOR/MONITOR'S ACRONYM(S)	
			11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited				
13. SUPPLEMENTARY NOTES The original document contains color images.				
14. ABSTRACT				
15. SUBJECT TERMS				
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES 14
a. REPORT unclassified	b. ABSTRACT unclassified	c. THIS PAGE unclassified		
19a. NAME OF RESPONSIBLE PERSON				

$T_{i,j,k}^m$	start time of assignment on target m by UAVs i, j, k
$t_{i,j,k}^{l,m}$	time the task performed by a UAV trio i, j, k on target m at stage l is completed
u_i	steering input for the i -th stand-in UAV
V	set of UAVs
v_d	minimum detection speed
v_i	speed of the i -th stand-in UAV
x_i	horizontal position of the i -th stand-in UAV
$x_{i,j,k}^{l,m}$	binary decision variable
y_i	vertical position of the i -th stand-in UAV
α	angular footprint of stand-in UAV sensor
ΔT	tracking duration
γ	offset angle from the heading of the moving target
Ω	target region of detectability
ω_{max}	maximum turning rate of the stand-in UAV
ϕ_i	heading of the i -th stand-in UAV
ψ	target heading angle

Abbreviations

CGMTE	cooperative ground moving target engagement
CMTAP	cooperative multiple task assignment problem
CTP	capacitated transhipment problem
GA	genetic algorithm
GAP	generalized assignment problem
GMTI	ground moving target indication
LOS	line of sight
MILP	mixed integer linear programming
SMTAP	simultaneous multiple task assignment problem
TSP	travelling salesman problem
UAV	unmanned aerial vehicle
VRP	vehicle routing problem

I. Introduction

MANY combinatorial optimization problems are concerned with pairing between agents and tasks. The simplest one is the classical assignment problem consisting of such optimal pairing, without assigning an agent more than once and ensuring that all tasks are completed. Such a problem can be easily solved by the Hungarian algorithm.¹ When an agent can be assigned to more than one task, or there are resource limitations to process a given task by an agent, the problem becomes much more complicated and cannot be solved in polynomial time. Such problems include the travelling salesman problem (TSP),² the generalized assignment problem (GAP),³ and the vehicle routing problem (VRP).⁴ In all of these classical problems the minimum cost assignment is sought, where: in the TSP, the tour is of one agent between a finite number of cities; in the GAP m agents need to perform n jobs, such that each job is assigned to exactly one agent and the resource of each agent is limited; and in the VRP m vehicles, with a given capacity, are dispatched from a single depot to deliver to n customers, each requiring a specified weight of goods, and then return to the depot.

In recent years a different class of problems denoted cooperative multiple task assignment problem (CMTAP) has been studied.⁵ This kind of problem arises for example when a group of unmanned aerial vehicles (UAVs) are assigned to perform multiple consecutive tasks such as classification, attack, and kill verification on multiple ground targets. For solving such problems, emerging algorithms of different classes have been proposed, including: mixed integer linear programming (MILP),^{6,7} iterative capacitated transhipment problem (CTP) algorithm,⁸ and iterative auction.^{9,10}

Stochastic search methods¹¹ might be considered in order to avoid the computational complexity of the combinatorial optimization methods described above and thus speed up the convergence to a good feasible solution. The genetic algorithm (GA) is such an approach that also does not require explicit computation of the gradients of the cost function.¹² Assuming that the search space is not extremely rugged,¹³ the GA will

often quickly yield good feasible solutions and will not be trapped in any local minimum; however, it may not yield the optimal solution. Another key attribute of the method is the possibility of parallel implementation. The application of GAs to classical combinatorial optimization problems such as TSP, GAP, and VRP has been widely studied.¹⁴⁻¹⁶ Recently, the GA methodology was used to solve the CMTAP showing promising results.⁵

A cooperative ground moving target engagement (CGMTE) was recently studied as a cooperative assignment problem using a MILP formulation.¹⁷ In the CGMTE scenario, several UAVs cooperatively track and attack mobile ground targets using doppler radars and GPS-guided weapons. This scenario features joint and overlapping tasks, where multiple vehicles have to perform coordinated actions simultaneously to successfully complete a mission. We denote this class of a problem as simultaneous multiple task assignment problem (SMTAP). In Ref. 17, the issues of path planning and task timing constraints were addressed using agent availability time windows in the MILP formulation. This method provides good solutions to the CGMTE problem, but due to the computational complexity can only be used for relatively small problem sizes. For more complex problems involving more than a few vehicles and targets, the solution time of the MILP becomes impractical for on-line implementation. Also, it requires for the cost function to be linear. In this paper, we use a similar process of calculating time windows and vehicle paths to determine assignment costs, but use a GA to efficiently search the space of possible solutions and provide the cooperative assignment. The GA also enables handling non linear cost functions, such as minimum group prosecution time.

The remainder of this manuscript is organized as follows: In the next section, the CGMTE is presented and associated tasks and UAV path planning issues are discussed. Then, the problem is posed as a new combinatorial optimization problem and its computational complexity is analyzed. This is followed by the derivation of a GA allowing efficient search of the solution space. A simulation section consisting of an analysis of a representative sample run and a Monte Carlo study is then presented. Concluding remarks are offered in the last section.

II. Cooperative Ground Moving Target Engagement

A. Scenario

In the CGMTE scenario, a stand-off UAV, with a wide-area ground moving target indication (GMTI) doppler radar, and several smaller stand-in UAVs, with small-area GMTI doppler radars and GPS-guided ground-attack weapons, must cooperatively track and prosecute moving ground targets. A very large number of potential targets could be simultaneously tracked by the stand-off vehicle, but we will assume that targets to be prosecuted are nominated to the UAV team by a human supervisor or outside agent. In order to meet the potentially severe timing constraints of such a scenario, the computation of an efficient set of task assignments and corresponding vehicle paths must be automated for practical implementation. Thus, we focus on the cooperative task planning required to track and attack the assigned targets.

The stand-off UAV is located outside of the region of interest and has a powerful GMTI doppler radar with an assumed 360-degree sensor footprint able to view the entire field of interest. For purposes of this assignment algorithm, the off-board vehicle is assumed to travel in a tight circular orbit such that its line-of-sight (LOS) to targets is approximately fixed. The vehicle can track any ground moving target within the region of interest, as long as the component of the target's velocity, relative to the terrain in the direction of the stand-off UAV, is above the required minimum detection speed v_d . For simplicity, we will assume that the stand-off UAV can provide continuous coverage of the region of interest, although gaps in coverage could be incorporated into the path planning algorithms. The stand-in UAV can similarly track ground targets using a GMTI doppler radar, with much smaller area coverage, and can also release a weapon on a target.

To reduce the position uncertainty each target must be simultaneously tracked by two vehicles with sufficient angular separation. Thus, during an attack a target inside the region of interest is tracked by a stand-in vehicle cooperating with either a stand-off vehicle or with another stand-in UAV. Let

$$V = \{0, 1, \dots, N_v\} \quad (1)$$

be the set of cooperating UAVs where 0 denotes the unarmed stand-off UAV and there are N_v stand-in UAVs carrying GPS-guided munitions. We assume that the vehicles fly at a constant speed, and that fuel

constraints are not a factor during a task assignment. Let

$$G = \{1, \dots, N_t\} \quad (2)$$

be the set of ground targets nominated to the UAV team for prosecution. This set of targets must be serviced by the $N_v + 1$ cooperating UAVs. For simplicity we assume that throughout the engagement the targets have a constant velocity, i.e. constant speed and heading. Let

$$S = \{1, 2, \dots, N_s\} \quad (3)$$

be the set of stages in which the assignment of three vehicles to each target is made. Since each target need be serviced by a UAV trio only once then $N_s = N_t$.

B. Tasks

The CGMTE requires that two or more UAVs with doppler radars track a moving ground target while an additional UAV launches a GPS-guided munition. The sensed target position and associated error ellipses from each tracking UAV are fused to form an accurate GPS location of the target, to which the munition is guided. In order to reduce the error covariance in the position estimate of the moving target, the UAVs tasked to perform the tracking must have sufficient LOS separation to the target, preferably near orthogonal views. Therefore, we require at least a 45 degree separation between the look angles of the two vehicles tracking a target, restricting the tracking regions and time windows. In addition, a moving target can only be detected and tracked by the doppler radars if the LOS view to the target is within some offset angle, γ , from the heading of the moving target, ψ . Thus, the region of detectability is

$$\Omega = \{\psi - \gamma, \psi + \gamma\} \cup \{\pi + \psi - \gamma, \pi + \psi + \gamma\} \quad (4)$$

Fig. 1 shows the heading of the target and the associated angular regions in which UAVs can be located to detect the target's motion.

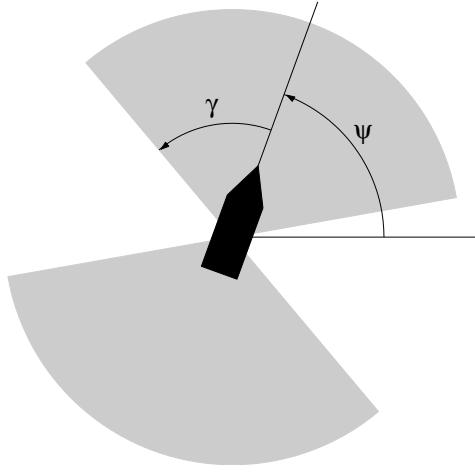


Figure 1: Region of detectability based on target heading

Complicating matters further, each stand-in UAV has a limited sensor footprint with minimum and maximum ranges (r_{min}, r_{max}), and a maximum angle away from the body y-axis (α) that the radar can view. Due to the configuration of the radar antenna array, the footprint is pointed out the wing of the UAV. Fig. 2 shows a stand-in UAV tracking a target and the associated sensor footprint relative to the orientation of the UAV. The sensor can scan on either side of the UAV, but not both at the same time. The dark area in this figure corresponds to the sensor footprint while the light grey area corresponds to the target's region of detectability (plotted in Fig. 1).

In total, each target $m \in G$ requires three overlapping tasks: two simultaneous tracking tasks, from two separate UAVs, and an attack task, performed by a third stand-in UAV. The simultaneous cooperative

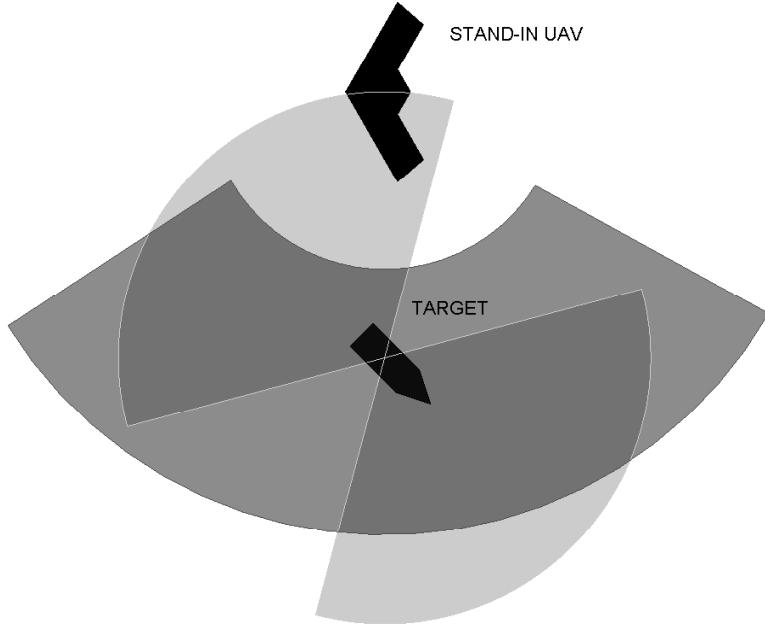


Figure 2: Sensor footprint (dark gray region) and detection cones (light gray region)

tracking tasks must begin before the weapon is launched, and continue until the weapon strikes the target. Thus, the tracking duration ΔT^m must satisfy

$$\Delta T^m \geq t_{flight} \quad (5)$$

where t_{flight} is the flight time of the weapon. For simplicity we assume that weapons are launched at a fixed distance from the target, resulting with a weapon flight time, t_{flight} , that is known and fixed.

C. Path Planning

Path optimization for the stand-in UAVs in the CGMTE is subject to numerous constraints. For this paper, we simplify the constraints by assuming that relative to the UAVs the targets are approximately fixed in position. We also restrict the cooperative tracking tasks to using a circular orbit with a pre-specified radius r_t . This path-planning strategy is suboptimal, but allows sufficient freedom to exercise the task assignment function. Improved path planning with fewer constraints is a subject of on-going research.

Although six degree-of-freedom dynamics are used in the vehicle simulation, for the purposes of path planning the UAV dynamics are modeled as a Dubin's car.¹⁸ Thus, UAV $i \in V \setminus \{0\}$ is assumed to fly with constant altitude and constant speed, and to have continuous-time kinematics given by:

$$\begin{cases} \dot{x}_i = v_i \cos \phi_i \\ \dot{y}_i = v_i \sin \phi_i \\ \dot{\phi}_i = \omega_{max} u_i \end{cases} \quad (6)$$

where (x_i, y_i) is the position of the UAV, v_i is the (constant) flight speed, ϕ_i is the heading, ω_{max} is the maximum turning rate, and u_i is the steering input where $|u_i| \leq 1$. The minimum turn radius is $R_{min} = v_i / \omega_{max}$.

Path planning is performed in several steps:

1. Calculate the set of all feasible times $T_{i,j,k}^m$ that vehicles $i, j, k \in V$ ($i \neq j \neq k$) can begin performing the specified tasks on target $m \in G$ for a duration of ΔT^m . This involves calculating the minimum time at which a vehicle can begin tracking the target, and adding loiter to the path before tracking begins if a delay is needed to satisfy timing constraints. For most vehicle and target initial conditions, any task start time that is later than the minimum one is feasible. However, due to the dynamic constraints of

the vehicle in some special cases continuous elongation of the vehicle path is not feasible.¹⁹ Determining the feasible time windows does not require calculation of a path for every possible start time.

2. Select the combination of time windows that allows minimum task completion time. Once the set of all possible times $T_{i,j,k}^m$ that vehicles $i, j, k \in V$ can perform the required tasks on target $m \in G$ have been calculated, we select the combination resulting in the earliest possible completion of target prosecution, while satisfying all constraints. Let $t_{i,j,k}^{l,m}$ denote the time the task, performed by a UAV trio, on target m at stage $l \in S$ is completed. For the fused estimate of the target position to have sufficient accuracy, the look angles of the tracking vehicles must be substantially separated (at least 45 degrees, as discussed earlier).
3. Calculate paths that meet the time windows specified in the previous step. Once specific start and stop times for each task are known, paths are calculated to meet those exact timing constraints. The vehicles' position at the end of these tasks, and the completion times, are saved and used as the new initial conditions for calculating vehicle paths for succeeding tasks.

III. Combinatorial Optimization Problem

In this section the CGMTE is posed as a combinatorial optimization problem and its computational complexity is analyzed.

A. Performance Criterion

We choose the performance criterion as the minimum time for the team to accomplish the required tasks on all targets. Thus, the group objective is to minimize the longest participation time of one (or more) of its members

$$J = \max_{i \in V} t_i > 0 \quad (7)$$

Let $x_{i,j,k}^{l,m} \in \{0, 1\}$ be a decision variable that is 1 if at stage $l \in S$ a trio of vehicles $i, j, k \in V, i \neq j \neq k$ performs the required tasks on target $m \in G$ and is 0 otherwise. This cost is computed using the path optimization algorithm discussed in the previous section. Note that for $l \geq 2$ computing $t_{i,j,k}^{l,m}$ might be dependent on the assignment at stage $l-1$ since some or all of the stand-in UAVs may have participated in previous tasks. We assume that the vehicles have no resource limitations, and thus can participate as much as needed in the group assignment. Thus, the cost function to be minimized is

$$J = \max \sum_{m=1}^{N_t} \sum_{i=0}^{N_v} \sum_{j=0}^{N_v} \sum_{k=1}^{N_v} t_{i,j,k}^{l,m} x_{i,j,k}^{l,m} \quad , \quad l \in S \quad (8)$$

B. Problem Constraints

The cost function of Eq. 8 is minimized under the set of constraints given next.

First, the decision variable $x_{i,j,k}^{l,m}$ must be binary

$$x_{i,j,k}^{l,m} \in \{0, 1\} \quad , \quad l \in S, \quad m \in G, \quad i, j, k \in V \quad (9)$$

Next, it must be ensured that at each stage exactly one target $m \in G$ is serviced by the UAV group. Thus,

$$\sum_{m=1}^{N_t} \sum_{i=0}^{N_v} \sum_{j=0}^{N_v} \sum_{k=1}^{N_v} x_{i,j,k}^{l,m} = 1 \quad , \quad l \in S \quad (10)$$

Third, at all stages each target $m \in G$ is serviced exactly once, resulting with

$$\sum_{l=1}^{N_s} \sum_{i=0}^{N_v} \sum_{j=0}^{N_v} \sum_{k=1}^{N_v} x_{i,j,k}^{l,m} = 1 \quad , \quad m \in G \quad (11)$$

Finally, it must be maintained that a vehicle is not assigned to multiple tasks on a target, i.e. three different vehicles service each target,

$$\sum_{i=0}^{N_v} \sum_{j=1}^{N_v} x_{i,i,j}^{l,m} + \sum_{i=0}^{N_v} \sum_{j=0}^{N_v} x_{i,j,i}^{l,m} + \sum_{i=0}^{N_v} \sum_{j=0}^{N_v} x_{i,j,j}^{l,m} = 0 \quad , \quad l \in S, \quad m \in G \quad (12)$$

C. Computational Complexity

Solving the combinatorial optimization problem posed above requires searching the space of feasible solutions. The computational complexity is mainly dependent on the number of possible solutions since the path planning, described in the previous section, is performed independently at each stage.

Theorem 1 n_f is the upper bound on the number of different feasible assignments where

$$n_f = [N_v^2(N_v - 1)/2]^{N_T} N_T! \quad ; \quad N_v \geq 2 \quad (13)$$

Proof: The number of target prosecution sequences is $N_T!$. The number of different trio combinations servicing each target is $N_v^2(N_v - 1)/2$. Since the assignment of a trio of vehicles is independent at each stage $l \in S$ and is independent from the target prosecution sequence then Eq. 13 is obtained.

Let different stand-in UAVs participate in two consecutive assignments. It is immediately evident that changing the order of these two tasks does not make any change to the group assignment. Thus n_f is an upper bound on the number of different group assignments. \square

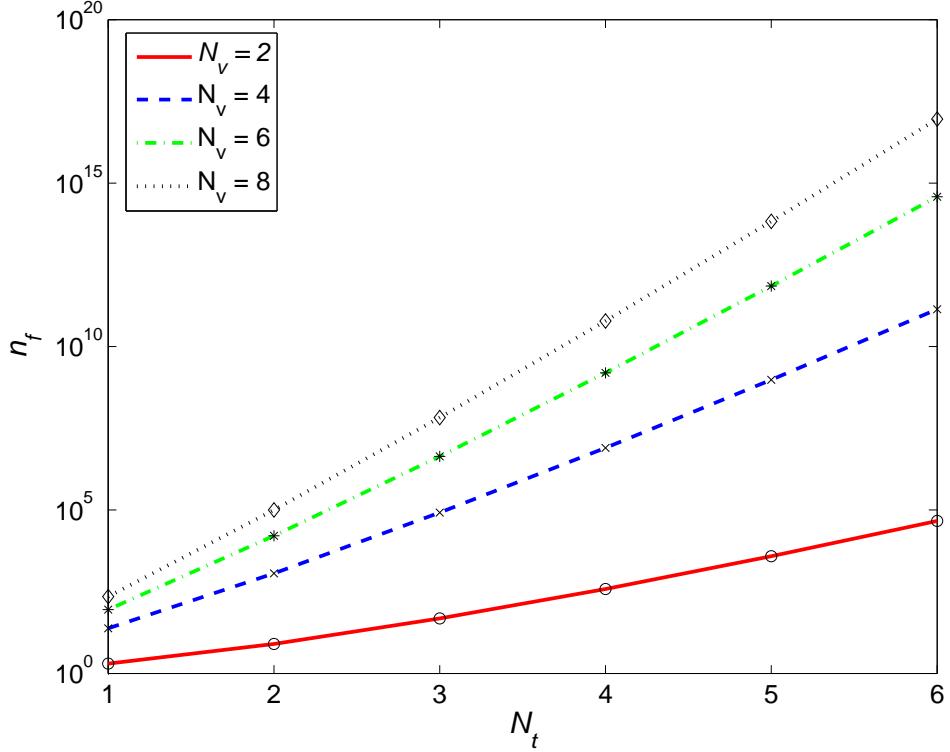


Figure 3: Upper bound on the number of assignments

The upper bound on the number of feasible assignments is plotted in Fig. 3 for different UAV team sizes. It is apparent that even for a relatively small number of targets and vehicles the number of feasible assignments is very large. Note that since the CGMTE is in the class of SMTAP it has a lower number of feasible assignments than a corresponding CMTAP (with 3 consecutive tasks per target) since the need to

perform tasks on a target simultaneously instead of consecutively prohibits a vehicle from servicing the same target more than once. However, the path planning and timing constraints are more complicated due to the severe coupling between the trajectories of the different UAVs.

IV. Genetic Algorithm

In this section we present a GA for solving the CGMTE, posed as a combinatorial optimization problem in the previous section. Due to the computational complexity of the problem, finding an optimal solution for this problem is intractable for relatively small size scenarios. The proposed GA enables obtaining good feasible solutions even for relatively large sized problems. The encoding of the GA chromosome is a major part of the solution process. The special encoding for the CGMTE is presented in the next section. Then, the application of the different genetic operators is discussed.

A. Encoding

An important part of applying a GA to an optimization problem is the solution encoding in the chromosome. Usually a string is employed for the encoding, while here for simplifying the encoding process and the application of the genetic operators we use a matrix.

Let

$$C = \{1, 2, \dots, N_c\} \quad (14)$$

be the set of each chromosomes' genes, where $N_c = 3N_t$. Thus, each chromosome matrix is composed of two rows and N_c columns corresponding to genes. Every 3 genes define a simultaneous task on one target. The first two of three genes in each set define the vehicles performing the cooperative tracking while the third gene defines the vehicle performing the attack by releasing a weapon.

An example chromosome is given in Fig. 4 where 6 vehicles (including the stand-off UAV denoted as vehicle 0) perform simultaneous tasks on 4 targets. The encoded assignment is as follows: vehicles 4 and 5 track target 2 until the weapon released by vehicle 2 hits it. Next, vehicle 2 together with the stand-off UAV track target 3 while vehicle 4 is responsible for the attack phase. At the same time target 1 is tracked by the stand-off UAV and vehicle 3 and attacked by vehicle 1. Then, vehicle 2 together with the stand-off UAV track target 4 until the weapon released by vehicle 1 hits it. Note that since the stand-off UAV can track multiple targets simultaneously then the tasks on targets 3 and 1 are independent. However, it can be easily seen that the tasks on targets 2,3, and 4 are dependent since vehicle 2 participates in all of them. Thus, these targets can not be prosecuted simultaneously, only in order.

Vehicle	5	4	2	0	2	4	3	0	1	0	2	1
Target	2	2	2	3	3	3	1	1	1	4	4	4

Figure 4: Example of chromosome representation.

B. Genetic Operators

We initialize the algorithm with N_f chromosomes and maintain a constant size solution population throughout the different stages. Off-line, the process of producing new generations can be repeated until some stopping criterion is met, e.g. the fitness of the best chromosome hasn't improved for a given number of iterations. For an on-line implementation the algorithm can be run for a specific allotted time. In this study the generations have been progressed for a given number of generations, denoted N_g . For creating a new population of solutions we employ the genetic operators: selection, crossover, mutation, and elitism.

In the selection stage two parent chromosomes are chosen from the population based on their fitness. Thus, the better the fitness, the higher is the probability of a chromosome to be reproduced to the next generation. As a selection tool we apply the Roulette wheel procedure. We evaluate the fitness of each chromosome solution using Eq. 8

$$f = 1/J \quad (15)$$

Crossover is performed at a single point across the chromosome, selected using a uniform distribution. In order to preserve the validity of the chromosome the crossover location is chosen in a quanta of 3 genes.

Thus, the first child solution consists of the first $3x \in C$ genes (columns) from the first parent and $N_c - 3x$ genes from the second parent; and vice versa for the second child. We apply this operator with a probability p_c .

The mutation operator, applied with a probability p_m , involves randomly changing one of the genes in the child chromosome. We mutate only the identity of one of the three vehicles participating in the simultaneous task. The identity of the new vehicle is selected randomly and we enforce that Eq. 12 is held, i.e. that three different vehicles service each target. The application of this operator prevents the algorithm from getting trapped in a local minimum.

In order to avoid the possibility of losing the best solutions, when propagating to the new generation, we employ the elitism genetic operator and keep the best N_e chromosomes from the previous generation. The rest of the new chromosomes ($N_f - N_e$) are obtained by repeatedly producing children, by the methods described above, until the new population is filled.

V. Simulation Study

In this section the performance of the GA algorithm in solving the CGMTE is analyzed using simulations. The area of interest where the targets and stand-in UAVs are located is 110km wide and 170km long. The simulation parameters are given in Table 1. In the next subsection a sample scenario consisting of 6 UAVs performing the simultaneous tasks on 3 targets is illustrated. Then, a Monte Carlo study is presented for comparing the results of the GA to a random search method for different sized problems. The sensitivity of the GA to tuning parameters is also investigated.

GA	Scenario	Sensor
$N_e = 4$	$N_v \in \{5, 10\}$	$r_{min} = 20\text{km}$
$N_f = 50$	$N_t \in \{3, 5\}$	$r_{max} = 50\text{km}$
$N_g = 50$	$v_d = 10\text{m/s}$	$\alpha = 60\text{deg}$
$p_m \in \{0, 0.033, 0.066, 0.1\}$	$r_t = 30\text{km}$	
$p_c \in \{0.7, 0.8, 0.9, 1\}$	$t_{flight} = 50\text{sec}$	

Table 1: Simulation parameters.

A. Sample Run

To illustrate the mission scenario discussed in this paper a sample scenario with 6 UAVs (1 stand-off and 5 stand-in UAVs) and 3 targets is examined. The location and heading of the targets and stand-in UAVs is plotted in Figs. 5 with the stand-off UAV located north of the region of interest. Fig. 5(a) shows the initial positions of the targets and stand-in UAVs. Each target is shown with an associated detectability region (outlined in black) and cooperative tracking region (solid wedge). The cooperative tracking region is the intersection of the detectability region with the LOS angles that have more than 45 degrees separation from the stand-off vehicle LOS. The assignment chromosome for this example is shown in Fig. 6.

As shown in Fig. 5(b), vehicle 5 cooperatively tracks target 3 with the assistance of the stand-off vehicle, while vehicle 3 performs the attack. Then, vehicle 2 cooperatively tracks target 1 with the assistance of the stand-off vehicle, while vehicle 4 performs the attack task, as illustrated in Fig. 5(c). Finally, as shown in Fig. 5(d), vehicles 1 and 5 track target 2 while vehicle 3 performs the attack. In this figure, vehicle 2 is still tracking target 1. Although vehicle 4 had released a weapon on target 1 in Fig. 5(c), that weapon is still in flight at the time of Fig. 5(d), so vehicle 2 must continue to track the moving ground target. The vehicles use loiter maneuvers as necessary to coordinate task timing and meet the required time windows. Loiter can be added by looping or adding turns to the flight path, as demonstrated by vehicles 1 and 5. In addition, before an attack task, a vehicle can loiter indefinitely (subject to fuel constraints) by circling a target just outside of the attack range, until the necessary cooperative tracking has been established. This behavior is demonstrated by vehicle 3, which exhibits a substantial period of loiter between attacking targets 3 and 2.

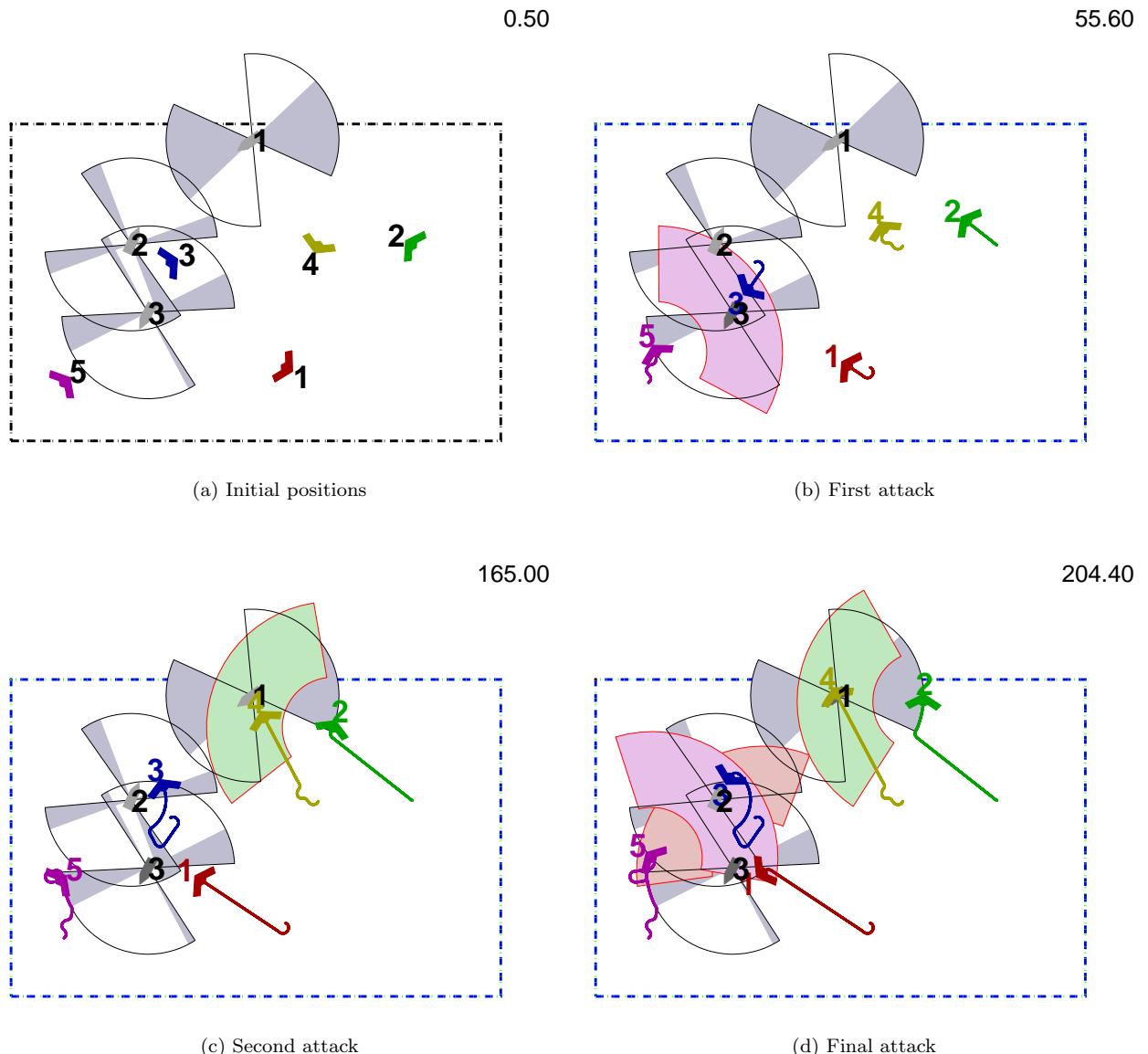


Figure 5: Simulated CGMTE.

Vehicle	0	5	3	0	2	4	1	5	3
Target	3	3	3	1	1	1	2	2	2

Figure 6: Example of chromosome representation.

B. Monte Carlo Simulations

A Monte Carlo study, consisting of 25 runs for each set of parameters, is used in this subsection to compare the performance of the GA to a random search algorithm. The random variables are the initial position and detection heading of the targets and the position and heading of the members of the UAV team. The first generation of GA chromosomes is composed of N_f random solutions, used to initialize the algorithm. We define J_1 as $\min J$ over this initial set of random solutions, where J is defined in Eq. 8. Note that J_1 is different for each set of initial conditions. To enable comparison, the results for each run are normalized by J_1 and an ensemble average is taken over the set of runs. Also, the same seed has been used to obtain the same initial conditions and N_f initial solutions for the sets of Monte Carlo runs. Note that in all the cases investigated the standard deviation of the solutions were in the same order of magnitude and small, thus validating that enough Mont Carlo runs have been performed.

In Figs. 7 and 8 the mean of J/J_1 is plotted as a function of the number of iterations for a scenario with $N_v = 5, N_t = 3$. The number of iterations corresponds to algorithm run time since it represents the number of feasible assignments being evaluated by the path optimization subroutine. Note that compared to the run time of the path optimization subroutine the overhead of applying the genetic operators is negligible. The results are plotted for the GA with different tuning parameters p_m and p_c ; and also for the random search algorithm. Note that due to the normalization method discussed above all the curves in the plots start at the point (50,1).

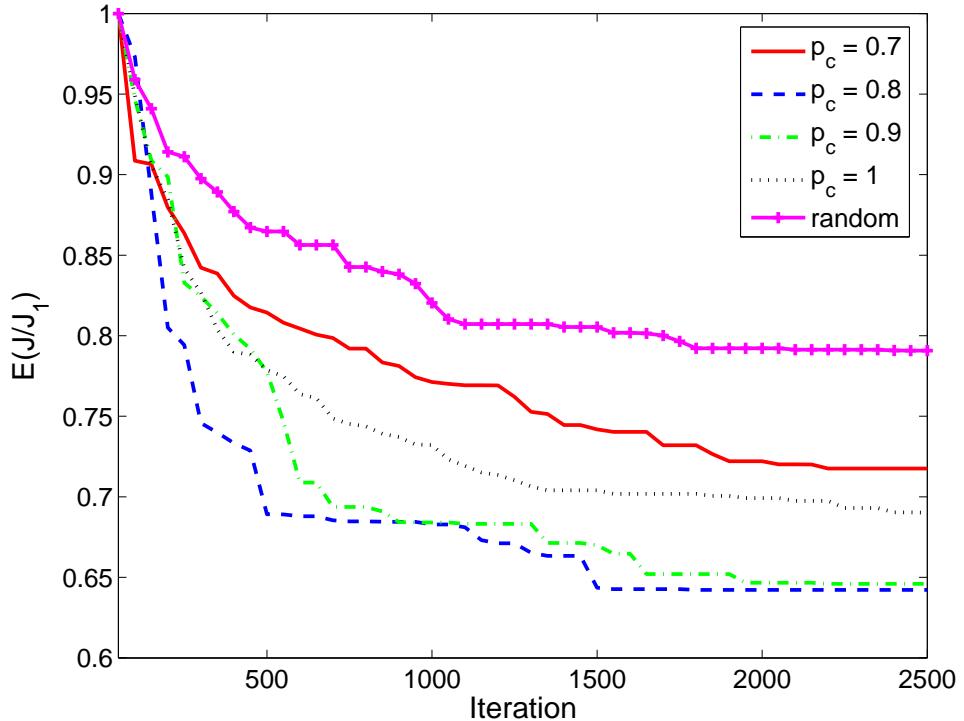


Figure 7: Solution quality comparison for $N_v = 5, N_t = 3$ with $p_m = 0.0666$

From Fig. 7, plotted for a mutation probability $p_m = 0.0666$, it is evident that the GA provides monotonically improving solutions that are considerably better than the random search method. The affect of changing the crossover probability is apparent and the best results are obtained for a crossover probability $p_c = 0.8$. Note that with a crossover probability that is too large ($p_c = 1$) or too small ($p_c = 0.7$) the quality of the results deteriorates. For $p_c = 1$ not enough good solutions mitigate as is to the next generation of solutions, while for $p_c = 0.7$ there are not enough perturbations around candidate solutions.

Plotting in Fig. 8 for $p_c = 0.8$ the results are obtained for different values of mutation probability p_m . It is apparent that if the mutation operator is not active, i.e. $p_m = 0$, then the GA quickly converges to solutions with a quality that is even worse than random search. This is caused since without the mutation operator the algorithm gets trapped in some local minimum. Using the mutation operator too often ($p_m = 0.1$) also

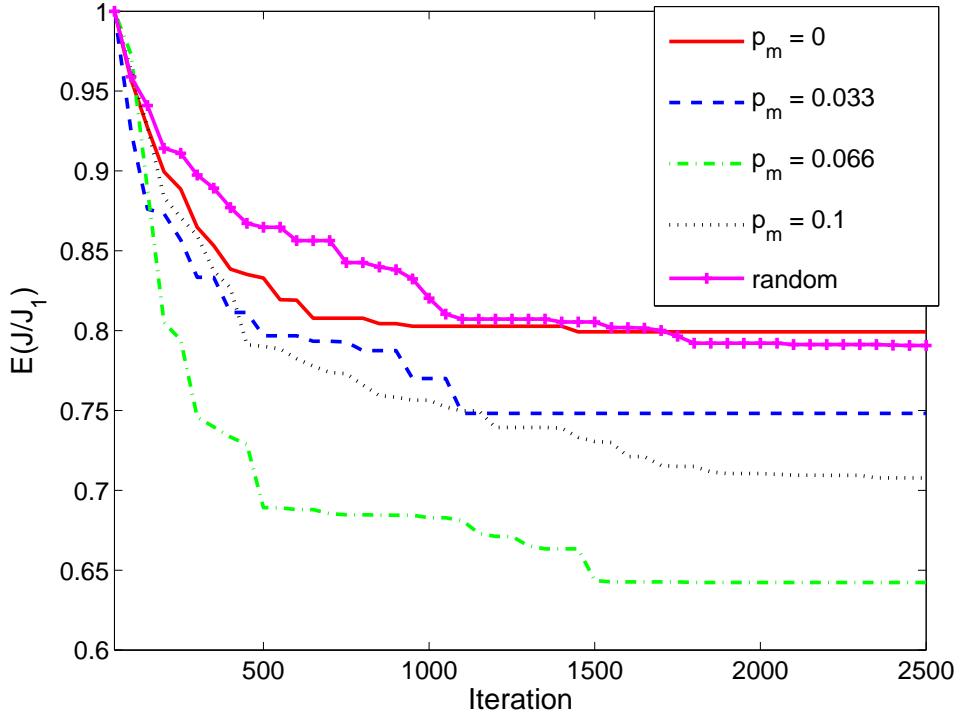


Figure 8: Solution quality comparison for $N_v = 5, N_t = 3$ with $p_c = 0.8$

deteriorates the results since too many perturbations are performed around good solutions.

In Fig. 9 the quality of the results, obtained with the GA after 2500 iterations, as a function of the tuning parameters p_m and p_c are plotted. The contours represent results of equal quality. It is apparent that the best results are obtained with $p_c = 0.8, p_m = 0.066$ and that for $p_m = 0.066, p_c \in \{0.8, 0.9\}$ the results are almost the same. Note that the coarse contours are due to the computational complexity of the problem restricting the number of data points. However, the qualitative sensitivity to these tuning parameters is clearly evident.

A similar comparison between the GA (with $p_c = 0.8, p_m = 0.066$) and random search is plotted in Fig. 10 for a scenario of with $N_v = 10, N_t = 5$. For comparison, the results for the previous scenario ($N_v = 5, N_t = 3$) are also plotted. Although the GA was not tuned for the larger sized scenario, it still substantially outperformed the random search algorithm. Note that such large sized scenarios are computationally infeasible for optimization methods such as MILP and thus optimal solutions can not be obtained. However, since the GA offers solutions that are monotonically improving over time then it can be used to obtain suboptimal solutions for a given allotted computation time.

VI. Conclusions

A new combinatorial optimization problem of assigning a group of UAVs to simultaneously perform tasks on multiple targets has been posed and solved. The scenario of interest consists of multiple ground moving targets prosecuted by a team of heterogeneous UAVs carrying designated sensors and/or weapons. Even for small sized scenarios, the problem has prohibitive computational complexity for classical combinatorial optimization methods due to timing constraints on the simultaneous tasks and the coupling between task assignment and path planning for each UAV. Using Monte Carlo simulation the performance of the GA was compared to random search for different sized scenarios and the sensitivity of the GA to tuning parameters was studied. It was shown that the GA can efficiently search the space of feasible solutions and substantially outperform random search. Since one of the main features of the GA is in providing monotonically improving solutions, it can be applied to large sized scenarios that are computationally intractable for other optimization methods such as MILP. Moreover, it enables dealing with nonlinear cost functions, such as minimum overall

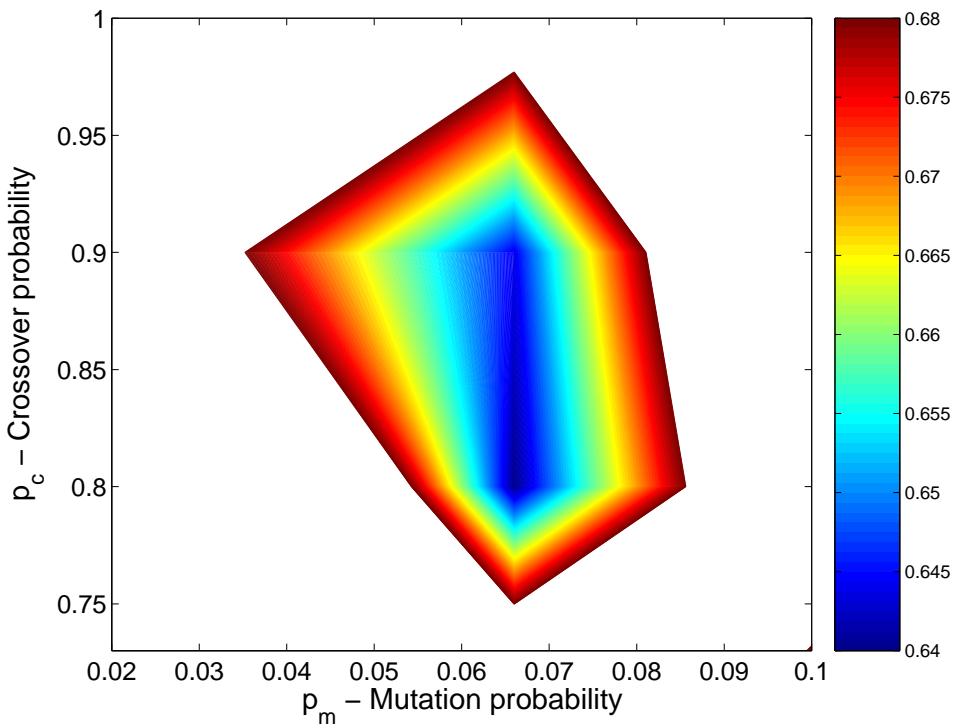


Figure 9: GA sensitivity to tuning parameters

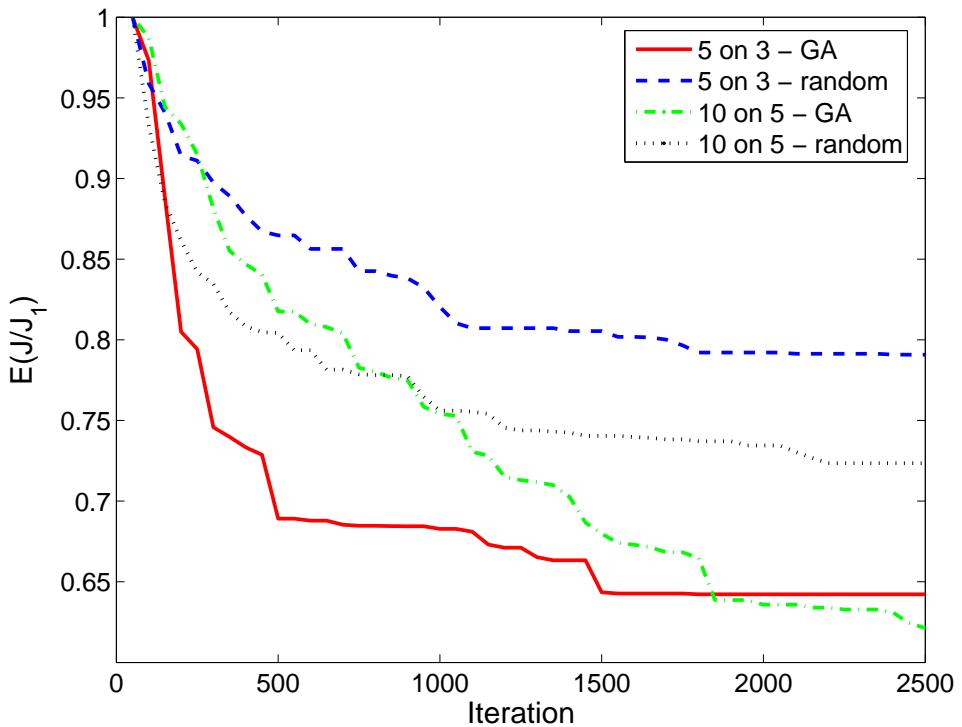


Figure 10: Solution quality for different sized scenarios

group prosecution time.

Acknowledgments

The authors would like to thank Derek Kingston and Steve Rasmussen for helpful discussions.

References

¹Kuhn, H. W., "The Hungarian Method for the Assignment Problem," *Naval Research Logistic Quarterly*, Vol. 2, 1955, pp. 83–97.

²Dantzig, G., Fulkerson, D., and Johnson, S., "Solution of a Large-scale Traveling Salesman Problem," *Operations Research*, Vol. 2, 1954, pp. 393–410.

³Catrysse, D. and Van Wassenhove, L., "A Survey of Algorithms for the Generalized Assignment Problem," *European Journal of Operations Research*, Vol. 60, 1992, pp. 260–272.

⁴Clark, G. and Wright, J., "Scheduling of Vehicles from a Central Depot to a Number of Delivery Points," *Operations Research*, Vol. 12, 1964, pp. 568–581.

⁵Shima, T., Rasmussen, J., Sparks, A., and Passino, K., "Multiple Task Assignments for Cooperating Uninhabited Aerial Vehicles using Genetic Algorithms," *to appear in Computers and Operations Research*, 2005.

⁶Richards, A., Bellingham, J., Tillerson, M., and How, J. P., "Coordination and Control of Multiple UAVs," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Monterey, CA, 2002.

⁷Schumacher, C. J., Chandler, P. R., Pachter, M., and Pachter, L., "Constrained Optimization for UAV Task Assignment," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Providence, RI, 2004.

⁸Schumacher, C. J., Chandler, P. R., and Rasmussen, S. J., "Task Allocation for Wide Area Search Munitions Via Iterative Network Flow Optimization," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, 2002.

⁹Bertsekas, D., "Auction Algorithms for network flow problems: A Tutorial Introduction," *Computational Optimization and Applications*, Vol. 1, 1992.

¹⁰Chandler, P. R., Pachter, M., Rasmussen, S., and Schumacher, C., "Multiple Task Assignment for a UAV Team," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, 2002.

¹¹Spall, C. J., *Introduction to Stochastic Search and Optimization*, John Wiley and Sons Inc., Hoboken, New Jersey, 1st ed., 2003, Wiley Interscience Series in Discrete Mathematics and Optimization.

¹²Mitchell, M., *An Introduction to Genetic Algorithms*, MIT, 1996.

¹³Bornholdt, S., "Genetic Algorithm Dynamics on a Rugged Landscape," *Physical Review E.*, Vol. 57, No. 5, 1998, pp. 3853–3860.

¹⁴Goldberg, D. E., *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, Massachusetts, 1989.

¹⁵Chu, P. C. and Beasley, J. E., "A Genetic Algorithm for the Generalised Assignment Problem," *Computers and Operations Research*, Vol. 24, 1997, pp. 17–23.

¹⁶Baker, B. M. and Aye chew, M. A., "A Genetic Algorithm for the Vehicle Routing Problem," *Computers and Operations Research*, Vol. 30, 2003, pp. 787–800.

¹⁷Kingston, D. and Schumacher, C., "Time-Dependent Cooperative Assignment," *Proceedings of the American Control Conference*, 2005.

¹⁸Dubins, L., "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal position," *American Journal of Math.*, Vol. 79, 1957, pp. 497–516.

¹⁹Schumacher, C., Chandler, Phillip R., R. S. J., and Walker, D., "Path Elongation for UAV Task Assignment," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Austin, TX, 2003.